

インターネット QoS ビジュアライザを用いた遅延とジッターの測定

福田 晴元 小野 諭 高橋 直久

NTTソフトウェア研究所

あらまし インターネットでは時間軸の保存の必要なアプリケーションが利用され始めており、このようなアプリケーションを品質よく実行するためには、遅延やジッターといった時間に関する品質をネットワーク全体について把握することが重要となる。本稿では、インターネットの品質情報を提供するインターネット QoS ビジュアライザを用いて、ネットワーク全体での遅延やジッターといった品質を把握する、パケットトラッキング手法について述べる。パケットトラッキングを行なうためには、ネットワーク上に複数の観測点を設け、観測点を通過したパケットのコピーと通過時刻を収集する。次に、収集されたパケットデータより、同一のパケットを捜し出してタイムスタンプを処理する。また、本稿では、インターネット QoS ビジュアライザを用いて、実験網で測定した遅延、ジッター、パケットロスについて示す。

The Measurement of Delay and Jitter using the Internet QoS visualizer

Harumoto FUKUDA Satoshi ONO Naohisa TAKAHASHI

NTT Software Laboratories

Abstract The Internet QoS Visualizer system provides quality information, such as delay or jitter, from an entire managed network. This quality information can be obtained by (1) stamping the time to packet data whenever the packet passes a probe point and sending a copy of the packet with the time-stamp to the center station as part of the packet stream from the probe point; (2) searching for the same packet in the many data streams from the probe points; and (3) analyzing the packet's time-stamp data from the probe streams. We discuss two measurement examples where the ability of a packet generator and Weighted Fair Queue method were used with the Internet QoS Visualizer.

1 はじめに

インターネットの普及にともない、時間軸の保存が必要な、音声や映像を扱うアプリケーションが利用され始めている。このようなアプリケーションを用いて、双方向での会話を品質よく行なうためには、遅延、遅延変動（ジッター）、パケットロスが少ないことが要求される。このように、時間に依存する品質を要求するアプリケーションの利用が進むにつれ、低遅延であり遅延変動が少なく、さらに低パケットロスといった品質が、インターネットに要求されはじめています。

このような要求に答えるため、インターネットでは、帯域保証 [1][2] や様々なキューイング方式 [3][4] を用いて、アプリケーションプロトコルやトラフィックの量に応じて品質を変更することが可能となってきた。また、アプリケーションがネットワークの品質に応じて何らかの制御を行なうことも可能となりつつある [5][6][7]。

これらの方法を利用することにより、ネットワークマネージャがネットワークの品質を改善することや、アプリケーションやアプリケーションのユーザがネットワークの品質に応じて適応的に制御することが可能となっている。

ところで、ネットワーク管理者が品質制御を行なうためには、ネットワークの品質状態をネットワーク管理者

に示すシステムが必要となる。しかし、現在のインターネットでは、ネットワークの管理者に対して、ネットワークの品質やその変動について、実時間で把握して提示することが困難である。これは、品質の変動箇所の把握とその品質状態について、実時間で細かく把握することが出来ないためである。

このため、時間軸の保存が必要となるアプリケーションが多数実行された際に、管理対象となるネットワークの遅延やジッターといった通信品質の様子や、品質低下箇所、品質変動の内容を、ネットワーク管理者が調べられない。つまり、ネットワークの品質制御を適応的に行なえないこととなる。

そこで、特定のアプリケーションが生成するトラフィックについて、そのトラフィックに関連するパケットを複数地点で監視し、ネットワーク上でのパケットの動作状態を実時間で把握することにより、品質低下や変動の状態と場所を知ることが重要となる。

さらに、上記のようなネットワークの運用状態を受動的に監視して把握することに加えて、管理対象となるネットワークの品質特性を、ネットワークの管理者が能動的に調べることもある。例えば、ある特定のアプリケーションに関連した品質低下箇所や、新しい品質制御方法の特性について、ネットワークの管理者が事前に調査測定し

て、その結果問題のある箇所が発見された場合には、そこに手を加えることが必要となる。

この際に、特定のアプリケーションに関連する試験パケットを生成し、そのパケットを観測することによって品質を把握することが必要となる。このために、パケット長やアプリケーションプロトコル番号といった各種パラメータを容易に設定可能とし、かつ、正確に一定のレートでパケットを生成する装置が必要となる。

我々はこのような背景から、インターネットの品質を監視してリアルタイムに表示するインターネット QoS ビジュアライザの研究を進めている [8]。本稿では、まず、インターネット QoS ビジュアライザの概要について述べる。次に、ネットワーク上でのパケットの動作を観測することにより、ネットワーク全体の品質把握を可能とするパケットトラッキング手法を提案する。

さらに、能動的に品質を把握するために必要となるパケット生成プログラムを作成しており、このプログラムについて、異なる OS を用いた際のパケット生成特性を測定し、評価を行なう。最後に、このパケット生成プログラムとインターネット QoS ビジュアライザを用いて、実験網で能動的な品質測定を行ない、その結果について述べる。

2 インターネット QoS ビジュアライザの概要

インターネット QoS ビジュアライザは、通過したパケットのコピーを作成し、そのパケットに対して時刻を刻印するパケット収集装置、パケット収集装置から情報を得て、タイムスタンプの処理を行なう集信サーバ、さらに、品質情報を表示する表示装置により構成される [8]。パケット収集装置は、ISDN のフレーミングクロックを利用した広域時計同期 [9] を用いて正確に時刻が同期されており、さらに、全ての時刻の進み方も同期している [10]。システムの構成を図 1 に示す。

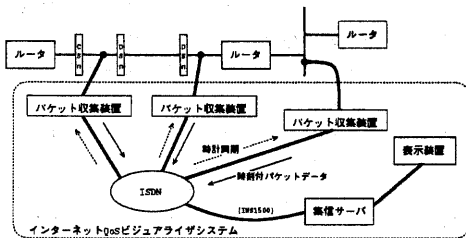


図 1 インターネット QoS ビジュアライザの構成

パケット収集装置は、10M イーサ、基本群、一時群、二次群のインタフェースを持ち、監視対象のネットワークを流れるパケットを覗くプローブを有する。パケット

収集装置がパケットをプローブする際には、BSD Packet Filter (BPF) [11] と呼ばれるパケットフィルタが利用される。さらにプローブされたパケットに対して、正確に同期した時刻情報をオンボードで刻印する。集信サーバより、パケット収集装置の BPF ヘプローブ命令を送ることにより、パケット収集装置より特定アプリケーションプロトコルのパケットを正確な時刻情報とともに収集出来る。

集信サーバでは、各パケット収集装置から送られてくるデータの処理手順を与えるプログラムが実行される。このプログラムを分析プログラムと呼び、データフロープログラムを用いて記述される。現在、分析プログラムの実行処理を行なうプログラムのプロトタイプ開発を行なっている。

3 パケットトラッキングによる品質評価

3.1 ネットワーク全体での品質把握に関する従来手法

従来より、NNstat や tcpdump [12] などのパケットをプローブするツールを用いてデータを時刻情報とともに収集している。この装置によりプローブされたパケットのヘッダを解析することにより、装置が設置された箇所の利用形態について統計的な情報が収集されている。例えば、一日や年間のトラフィック変動や、どのようなサービスを利用したパケットが時間帯によってどの程度流れていたか等の、ネットワークの利用形態についての情報が収集されている [13]。

上記のように、従来のトラフィック解析では、プローブの置かれた箇所でのパケット情報を収集して解析することに重点がおかれており、複数点でプローブした情報を比較して解析することは行なわれていない。このため、例えば、遅延やジッターのような、2点での情報を比較する必要のある品質情報は収集されていない。また、2点でプローブしたとしても、各収集装置の時計が正確に揃えられないため、2点での情報を比較して正確に通信品質を求めることは困難である。

3.2 パケットトラッキングの概要

インターネット QoS ビジュアライザでは、プローブ装置を複数台用意し、タイムスタンプとともにパケットを収集し、収集したタイムスタンプとパケットの情報を比較することにより、ネットワーク全体での遅延やジッター等の通信品質を測定する。具体的には、以下に例示するように、複数の観測点において監視対象となるパケットの通過経路と通過時間を把握することによりアプリケーション単位でのネットワークの品質把握を行なう。このように、特定のパケットに注目し、そのパケットを追跡して詳細な通信品質を求める方法をパケットトラッキングと呼ぶ。以下では、パケットトラッキングにより、遅延とジッターを求める方法を例示する。

- (1) バケット収集装置を複数台ネットワーク上に設置する。ここでは例のために、4台、つまり、 $P1, P2, P3, P4$ を設置したとする。
- (2) 監視対象のバケットがバケット収集装置を通過した際に、通過時刻のタイムスタンプとバケットのコピーを収集する。例えば、バケット p が、 $P1$ にてプローブされた時刻を T とすると、このデータを、 $P1(T, p)$ と表す。
- (3) バケット収集装置により集められたバケットデータより、同一のバケットを捜し出す。つまり、 $P1(T1, p), P2(T2, p), P3(T2, p), P4(T4, p)$ なるデータを探し出す。
- (4) プローブを通過する順序と、各プローブ間での遅延を求める。時刻を昇順に並べた結果が $T1 < T2 < T4 < T3$ である場合には、バケットが $P1, P2, P4, P3$ の順に通過したと把握できる。さらに、 $D1 = T2 - T1, D2 = T4 - T2, D3 = T3 - T4$ を求めることにより、観測点間での遅延を把握出来る。
- (5) バケット p が、アプリケーション A のトラフィックである場合には、アプリケーション A の遅延の統計値として、 $D1, D2, D3$ を用いる。さらに、 $D1, D2, D3$ を用いてアプリケーション A のジッターを求める。

3.3 バケットトラッキングの実現法

2節で述べたように、インターネット QoS ビジューライザは、ネットワーク上に複数のバケット収集装置を設置し、監視対象となるバケットがバケット収集装置を通過した際に、その通過時刻とバケットのコピーを収集する。さらに、プローブにて収集されたバケットのデータを集信サーバに送ることが可能である。また、正確な時計同期機構を利用して、遅延やジッターといった2点間での時刻情報をもとにした品質把握が可能である。これらの機能に加えて、以下の機能により、インターネット QoS ビジューライザではバケットトラッキングを実現する。

- **バケットマッチング** 集信サーバにおいて、複数のバケット収集装置から連続して送られて来るバケットデータの一部を参照して、同一のバケットを捜し出す。
- **統計処理** バケットマッチングにより生成される同一バケットのデータからバケットの通った経路を求める。さらに、時刻情報に注目して、遅延やジッターといった品質情報をアプリケーションのトラフィックフロー単位で作成し、トラフィックフロー別に品質情報を統計的に常時更新する。
- **表示** 測定されたトラフィックフローの伝送経路と品質情報をもとにして、ネットワーク全体でのトラフィックフロー別の動作と品質情報を実時間

で図示する。

4 バケット生成プログラムの特性評価実験

4.1 プログラムの機能概要

ネットワークの品質を事前に測定するといった、能動的な品質測定を行なうためには、バケットを柔軟に生成する装置が必要となる。このため、UNIX のソケットを利用してバケットを送出するバケット発生プログラムを作成した。これは、ualarm を利用して定期的にシグナルを発生させ、バケット生成プログラムがシグナルを受けとった際に、バケットを送出するプログラムである。本プログラムでは、バケットスペース (usec/pkt)、ペイロード長 (octets)、アプリケーションプロトコル番号、生成時間を指定することが出来る。

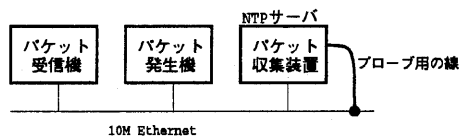


図2 実験網の構成

次に、図2にあるような実験網を構成して、バケットの生成状態を観測する。バケット収集装置が持つ時計は正確に進む [10] ため、試験バケットを時刻情報と共にプローブすることによって、バケットの生成状態を把握することが出来る。さらに、バケット収集装置は、Network Time Protocol (NTP) [14] のサーバとしても動作しており、イーサネットに IP 接続されている。バケット発生装置は、時刻の進み方を表すカーネル変数を調整した上で、NTP クライアントとして動作させて時刻同期を行なっている。

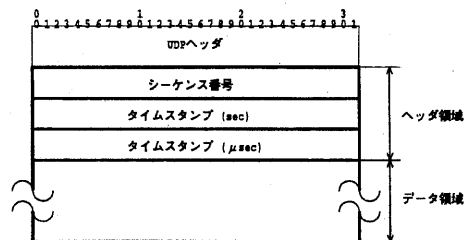


図3 UDPパケットの構造

バケット発生装置には、OS "A" (SYSVR4 系) と OS "B" (BSD4.2 系) の、異なる二つの OS を使い、バケット生成プログラムを実行する。実験では、生成されるバケットのデータ領域を 320 octet とする。また、43 msec/packet 間隔でバケットを生成するように実行し、連続して 150 分間生成する。

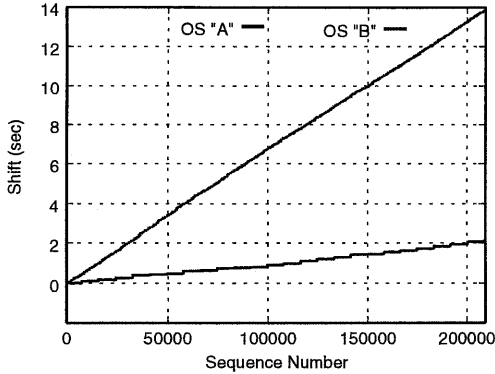


図4 想定したプローブ時刻と実測値の差 (43msec)

表1 パケットの生成状況

OS	発生時間 (sec)	送信パケット数 (pkt)	送信レート (msec/pkt)
"A"	9000.801158	209272	43.010059
"B"	9000.546806	208994	43.066053

利用したパケット生成プログラムは、図3に示すようなヘッダとデータ部分からなるUDPペイロードを持つようなパケットを生成する。なお、この場合には、ペイロード長とはデータ部分の長さを指定することとなる。シーケンス番号は0で始まり、パケットが生成されるとインクリメントされる。タイムスタンプは、UNIXのtimeval構造体を用いており、データをソケットに送る直前に時刻が埋め込まれる。

4.2 特性評価実験

パケット生成プログラムがパケットを生成する予想タイミングと、実際にネットワーク上でプローブされた時刻との差を示す。図4は、OS "A"とOS "B"上でパケット発生プログラムを動作させた結果を示す。図の横軸はUDPパケットに埋め込まれるシーケンス番号を示す。図の縦軸は、実際にプローブされた時刻から、パケットがプローブされる予想時刻を引いた値を示している。なお、プローブされる予想時刻は、シーケンス番号を Seq 、最初に生成されたパケット(シーケンス番号0のパケット)がプローブされた時刻を T とすると、 $0.043 \times Seq + T(\text{sec})$ とする。さらに、図4の一部を拡大したものを、図5に示す。

次に、図6は、UDPパケットに埋め込まれるタイムスタンプとパケット収集装置によりプローブされた時間との差分を示す。なお、この測定では表1に示すようにパケットが生成されたことを確認した。

図6に見られるように、両システムの時計は、150分の間においてmsec以下の範囲で正確に進んでいると考

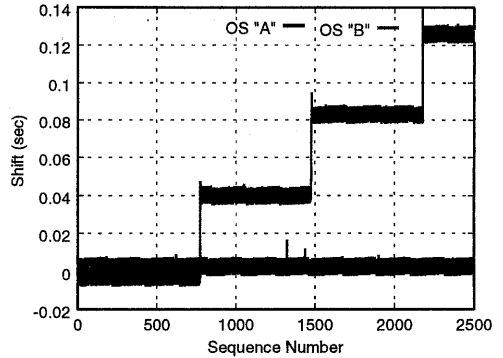


図5 想定したプローブ時刻と実測値の差 (拡大図)

えられる。また、図5に見られるように、両システムとも、10msecの幅で前後しながら、パケットを生成していることがわかる。ところで、図4にあるように、OS "B"は、予想されるプローブ時刻より、実際のプローブ時刻が遅れてきており、シグナルの発生タイミングの累積誤差が大きいたことがわかる。システム時計はmsec以下の精度をもち、さらに、発生タイミングのずれは約40msecである(図5)ことより、この累積誤差はシグナルのロスによるものではないかと考えられる。このように、OS "B"は、OS "A"に比べてualarmに与えるインターバルタイムの解像度が低く、OS "A"の方が、より定期的にパケットを生成できるのではないかと考えられる。このため、OS "A"を用いて本プログラムを実行する場合には、OS "B"に比べて、正確な一定のレートでパケットを生成出来る。なお、次節以降の測定では、累積誤差の少ないOS "A"を用いて試験パケットを生成する。

5 WFQの特性評価実験

前節で述べたパケット生成プログラムを用いて、実験網上で遅延とジッターの測定を行なった。実験網の構成を図7に示す。実験網では、パケット収集装置a, bを用意し、この二つの装置でパケットをプローブした際のタイムスタンプを比較することにより、遅延とジッターを測定する。

ルータ間のリンクのクロックは、約125Kbpsとし、試験パケットにより、2種類のトラフィックを生成した。これらは、サービスポート番号が異なり、図3の構造を持つUDPパケットである。これらを、Traffic A, Traffic Bと呼ぶ。Traffic Aは、UDPパケットのデータ部分が512 octetであり、10 packet/secの間隔で60秒間生成される。Traffic Bは、UDPパケットのデータ部分が512 octetであり、25 packet/secの間隔で30秒間生成される。なお、Traffic Bは、Traffic Aの

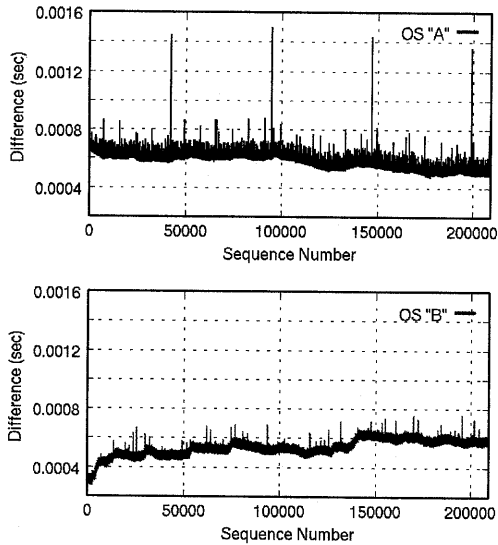


図6 パケット送信時刻とプローブ時刻の差

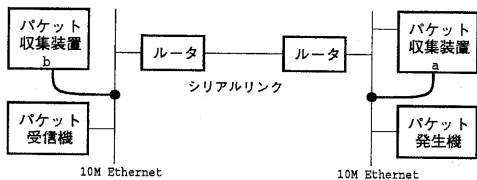


図7 遅延とジッターの測定実験網

生成が開始された後、10秒後に生成される。

測定では、ルータにおいてWeighted Fair Queuing (WFQ)[4][3][16]を用いた。これにより、転送レートが大きいトラフィックに関しては、コントロールアクションが発生し、遅延が増大してパケットロスが発生すると予想される。

パケット収集装置 a, b によりパケット i がプローブされた際の時刻を T_{a_i}, T_{b_i} とすると、パケット i の遅延 d_i は、

$$d_i = T_{b_i} - T_{a_i}$$

として求まる。図8に測定を開始してからの、両トラフィックの遅延 d を示す。

次に、パケットスペースの変化をジッターとしてとらえると、以下に示す D を用いて、 $D(i-1, i)$ をジッターとして定義可能である。

$$D(i, j) = (T_{b_j} - T_{b_i}) - (T_{a_j} - T_{a_i}) = d_j - d_i$$

しかしここでは、文献[15]で示されている方法を利用する。これは、 $1/16$ (4 bit shift) というゲインパラメー

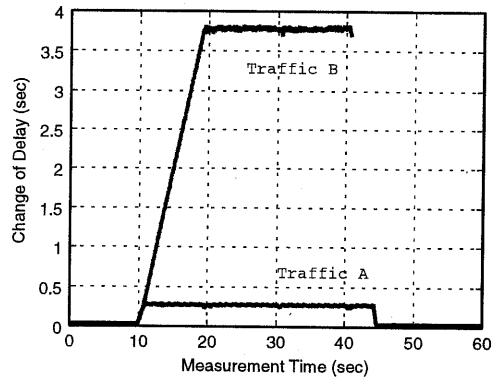


図8 WFQにおける遅延の測定結果

タによりノイズを取り去った結果をジッターとして用いる方法であり、ジッター J を以下の式で算出する。

$$J \leftarrow J + (|D(i-1, i)| - J) / 16$$

上記計算により得られるジッターを図9に示す。

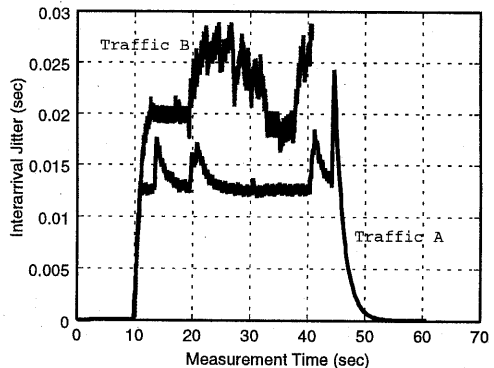


図9 WFQにおけるジッターの測定結果

このような状況において、それぞれのトラフィックにおけるパケットロスの累積を図10に示す。

WFQを用いた場合には、図8に示すように、Traffic Aの遅延はTraffic Bに対して小さな値となっている。また、図9を見ると、Traffic Bのジッターに比べると、Traffic Aのジッターは、小さくおさえられている。さらに、パケットロスに関しては、Traffic Aはロスがなく、これは、WFQがキューへのスケジューリングに利用する平均転送レート (AR) の値を、Traffic Aの転送レートが越えていないことを示す。このように、Traffic Aのような転送量の小さなトラフィックの品質は、Traffic Bのような輻輳が発生するパスト的トラフィックの発生の影響を比較的受けていないことがわか

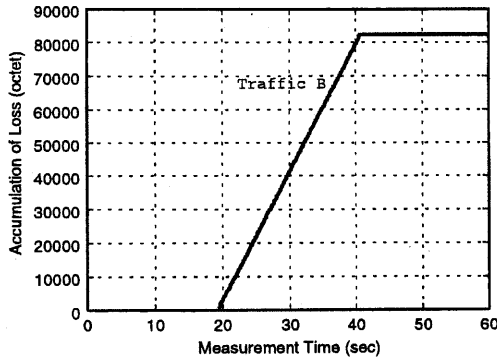


図 10 WFQ におけるパケットロスの累積

る。

インターネット QoS ビジューライザを用いて、このような情報を、ネットワークの複数のポイントから集めて提供することにより、ネットワーク全体の遅延やジッターといった品質情報が把握可能となる。なお、二つの装置間のパケットブロープ時刻の差分は数百 μsec 以内の幅で収まる [8] ため、この時刻をもとにした品質の測定を正確に行なえる。このようにして、ネットワーク全体の品質情報を把握することにより、ネットワーク管理者に対して、何らかの品質改善を施す動機を与えることが可能となると考えられる。さらに、ネットワーク管理者の対策の結果を正確に確認することも可能となる。

例えば、ある重要なトラフィックに関して、品質を観測した結果、悪影響があったと観測されたとする。これは、この状況を改善するという動機を与えたこととなる。さらに、対策としてネットワークのコントロールを変更した場合、例えば、WFQ を採用した場合には、今回の測定のように、遅延とパケットロスへの有効性を確認できる。

6 おわりに

本稿では、インターネット QoS ビジューライザを用いたパケットトラッキング方法について述べた。これは、複数の測定点を設置し、そこを通過したパケットのコピーを作成して通過した時刻を刻印する。その情報からパケットマッチングを行なうことにより同一パケットを探し出し、パケットの通過経路と通過時刻を求める。最後に、時刻情報の処理を行ない、そのパケットを発生したアプリケーションの情報を更新し表示を行なう方法である。また、本稿では、パケット生成プログラムの特性と WFQ を用いた際の遅延、ジッター、パケットロスからなる品質情報を、インターネット QoS ビジューライザを用いて、測定した。

今後は、収集したパケットについて、分析プログラム

の記述に従って、パケットマッチングを行なうシステムを構築していく予定である。また、本ビジューライザを実際のインターネット上で利用をすすめていき、ネットワークの品質測定を進める予定である。

最後に、本研究の御支援、御支援頂く市川晴久広域コンピューティング研究部長ならびに、日頃御討論頂く宮田輝子さん、ならびに、超並列コンピューティング研究グループの皆様へ深謝します。

参考文献

- [1] Zhang, L., Deering, S., Estrin, D., Shenker, S., and D. Zappala, "RSVP: A New Resource ReSerVation Protocol", IEEE Network, September 1993.
- [2] R. Braden, Ed, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP)", Internet Draft, Jun, 1997.
- [3] P. Goransson, "Bandwidth reservation on a commercial router", Computer Networks and ISDN System 28, pp.351-370, 1996.
- [4] C Huitema, "Routing in the Internet", Prentice Hall PTR, ISBN0-13-132192-7, 1995.
- [5] J-C Bolot, T Turletti, "Adaptive Error Control for Packet Video in the Internet", Proc. ICIP'96, Sep. 1996.
- [6] S MacCane, V Jacobson, "Receiver-driven Layered Multicast", SIGCOMM'96, pp.117-130, Aug. 1996.
- [7] S McCane, M Vetterli, "Joint Source/Channel Coding for Multicast Packet Video", Proc. ICIP'95 Oct. 1995.
- [8] 福田晴元, 小野諭, 高橋直久, "インターネット QoS ビジューライザの設計と実現" 信学会論文誌 B-I, Vol. J80-B-I, No.6, pp.438-446, Jun 1997.
- [9] 山下高正, 小野諭, "ISDN 網を用いた分散高精度時刻/周波数同期", 情報研報 Vol.95, No.61, pp.37-42, Jul. 1995.
- [10] 山下高正, 小野諭, "高速デジタル網を用いたクロック周波数同期", 信学会論文誌, Vol.J79-D-I, No.8, pp.522-529, Aug. 1996.
- [11] S. McCane, Van Jacobson, "The BSD packet filter: a new architecture for user-level packet capture", Winter USENIX, Jan. 1993.
- [12] Daniel C. Lynch, Marshall T. Rose, "Internet system handbook", ADDISON-WESLEY PUBLISHING COMPANY INC., Jan. 1993.
- [13] K C. Claffy, G C. Polyzos, and H-W Braun, "Traffic characteristics of the T1 NSFNET backbone", Proc. IEEE Infocom 93, Vol. 2, pp.885-892.
- [14] David L. Mills, "On the Accuracy and Stability of Clocks Synchronized by the Network Time Protocol in the Internet System", Computer Communication Review, Vol.20, No.1, pp.65-75, Jan. 1990.
- [15] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC1889, Jan. 1996.
- [16] L Zhang, "VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks", SIGCOMM'90, pp.19 - 29, 1990.